

AD-A184 843

END OF YEAR TECHNICAL REPORT DYNAMIC IMAGE
INTERPRETATION FOR AUTONOMOUS (U) MASSACHUSETTS UNIV
AMHERST DEPT OF COMPUTER AND INFORMATION S

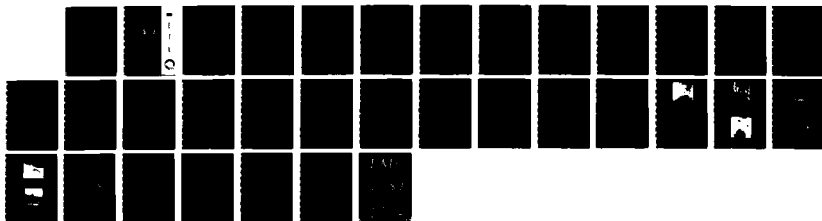
1/1

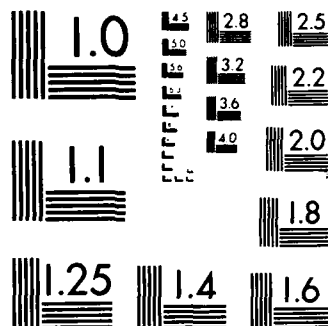
UNCLASSIFIED

E M RISEMAN ET AL MAY 87 ETL-8463

F/G 17/7 1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ETL-0463

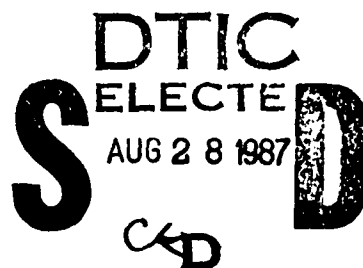
DTIC FILE COPY

2

AD-A184 043

End of year technical report:
Dynamic image interpreta-
tion for autonomous vehicle
navigation

E. M. Riseman
A. R. Hanson
L. J. Kitchen



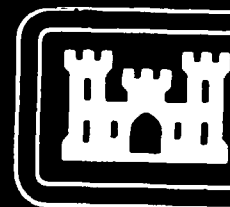
Computer & Information Science Department
University of Massachusetts
Amherst, MA 01003

May 1987

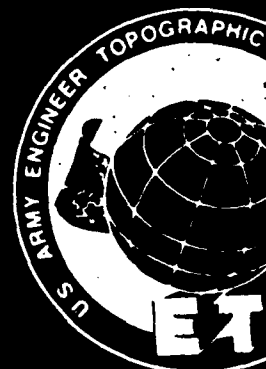
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

87 8 28 009



E
T
L



Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official
Department of the Army position unless so designated by other
authorized documents.

The citation in this report of trade names of commercially available
products does not constitute official endorsement or approval of the
use of such products.

8a. NAME OF FUNDING / SPONSORING ORGANIZATION DARPA	8b. OFFICE SYMBOL (If applicable) ISTO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DACA76-85-C-0008	
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, Virginia 22209-2308		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO. 62301E	PROJECT NO. TASK NO. WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) End of Year Technical Report: Dynamic Image Interpretation for Autonomous Vehicle Navigation			
12. PERSONAL AUTHOR(S) Riseman, E.M., Hanson, A.R., and Kitchen, L.J.			
13a. TYPE OF REPORT Annual	13b. TIME COVERED FROM 2/26/86 TO 2/25/87	14. DATE OF REPORT (Year, Month, Day) 1987 May	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION			

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Scene interpretation Sensor motion Spatial reasoning
FIELD	GROUP	SUB-GROUP	
17	07		
17	08		

End of Year Technical Report: Dynamic Image Interpretation for Autonomous Vehicle Navigation

Contract: DACA76-85-C-0008
February 26, 1986 - February 25, 1987

E. M. Riseman
A. R. Hanson
L. J. Kitchen

Computer and Information Science Department
University of Massachusetts
Amherst, MA 01003

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



Abstract

This report describes work conducted at the University of Massachusetts for the Autonomous Land Vehicle Project (under contract DACA76-85-C-0008) during the one-year period from February 26, 1986 to February 25, 1987. In pursuit of the goal of achieving dynamic image interpretation for autonomous vehicle navigation, we have made significant progress in the knowledge-based interpretation of road scenes, in visual motion analysis, and in mobile robot navigation. This work has been supported by development of necessary software tools, installation of appropriate hardware, and concurrent investigations into applicable techniques for image analysis.

Contents

1	Introduction	1
2	Knowledge-Based Image Interpretation	1
2.1	The Schema System	2
2.2	Road-Scene Interpretation	3
2.3	The Schema Shell	3
2.4	Intermediate Symbolic Representation	3
2.5	Object Hypothesis System	4
2.6	Grouping and Perceptual Organization	5
2.7	Goal-Directed Intermediate-Level Executive	7
2.8	Low-Level Vision System	7
3	Visual Motion Analysis	7
3.1	Recovery of Motion Parameters	8
3.2	Refinement of Depth Maps	8
3.3	Analysis of Frame-to-Frame Correspondence	9
3.4	Software for Motion Analysis	9
4	Mobile Robot Navigation	10
4.1	Autonomous Robot Architecture	10
4.2	Vision Modules	12
4.3	Robot Navigation Runs	12
	References	13
	Figures	17

1 Introduction

Research at the University of Massachusetts for the Autonomous Land Vehicle (ALV) Project is concerned with problems in dynamic image interpretation for autonomous navigation. In particular, our work has the following long-term research goals relevant to Task D of the ALV Project:

1. Determine the motion parameters of a sensor relative to the static environment.
2. Distinguish moving objects from the static environment and determine their motion parameters.
3. Develop algorithms for tracking and predicting the motion and environmental location of the sensor and moving objects.
4. Build a reliable depth map of the environment from combined motion, stereo, and laser range data.
5. Identify major objects (both static and moving) in the environment while the sensor is either stationary or in motion.
6. Interpretation of the environment (i.e., object identification in road scenes) to provide constraints for identifying and tracking moving objects.
7. Provide information to update an environmental model of the moving sensor, including location of the sensor, other moving objects and distinguished stationary objects.
8. Provide control information to an expert navigational and spatial-reasoning system for the purposes of path planning and obstacle avoidance.
9. Integrate all of the above capabilities into a flexible and extensible system for dynamic scene interpretation.

In the following sections of this report, we will present our work on the ALV project in the following areas: knowledge-based image interpretation, visual motion analysis, 3D object recognition, surface reconstruction, and mobile robot navigation.

2 Knowledge-Based Image Interpretation

The past year has seen the refinement of our concepts of schema-directed, knowledge-based image interpretation, the development of a number of tools and techniques, and the application of these concepts, tools, and techniques to the interpretation of a number of images of road scenes.

2.1 The Schema System

A central problem in image understanding is the representation and use of available sources of domain knowledge during the interpretation process. Each of the many different kinds of knowledge that may be relevant during the interpretation process imposes different kinds of constraints on the underlying representation and may lead to very different kinds of strategies for its effective use. Over the past several years, we have developed the notion of a 'schema' as the basic unit of knowledge representation in the VISIONS system. Within the schema system, image interpretation is the process of instantiating a subset of schemas to build a description of the three-dimensional scene which gave rise to the image. Knowledge is represented in an abstraction hierarchy of schema nodes by part/subpart descriptions, class/subclass descriptions, and expected relationships between schemas; the resultant hierarchical graph constitutes the VISIONS knowledge network. This approach has evolved over a considerable period of time, with recent work documented in [15,19,37].

Each schema node may be viewed as a 'packet' of information related to the recognition of an instance of a particular object class in an image, including properties and relations of extracted image events as well as control information expressed in the form of interpretation strategies. One or more of these strategies are executed when a schema is instantiated (i.e. when a copy is activated), to process a specific area of the image. Schema activation may be either bottom-up, where image descriptions imply the potential relevance of a schema, or top-down as the result of the context of a partial interpretation written on a blackboard communication structure by other schemas. Many schemas may be active at any one time, and the interpretation strategies provide control over the parallel interpretation processes and make use of a set of object-independent processes called knowledge sources. In our system, schemas communicate indirectly by posting object hypotheses on a blackboard.

The system is organized around three levels of data representation and types of processing. At the low-level, the representations are in the form of numerical arrays of sensory data with processes for extracting the image events that will form the intermediate representation. At the intermediate level, the representation is composed of symbolic tokens representing regions, lines, surfaces and their attributes (which might include local motion and depth information). At the high level, the representation is a set of object hypotheses and active schema instances which control the intermediate and low-level processes. Control initially proceeds in a data-directed manner and later is significantly top-down in a knowledge-directed manner.

Based on our experience with an initial implementation of the schema system and a set of experiments designed to interpret reasonably complex house scenes [8,9,18,19,28,37], a new schema system and support environment has been implemented and used for interpretation of road-scene images [15]. This and related work supported by this contract will be described below.

2.2 Road-Scene Interpretation

As a demonstration of the capabilities of the schema approach, a knowledge base of schemas and their associated strategies was created for doing image interpretation in the road-scene domain. The objects included in the knowledge base were: *sky, foliage, tree trunk, road, roadline, roadside gravel, telephone pole, telephone wire, roof, stop sign, yellow (caution) sign*, and *road scene* itself. The results of using this schema system for the interpretation of a number of complex road-scene images are described more fully in [15]. What is remarkable about this achievement is that once the supporting tools were in place, the development of the knowledge base took only a relatively modest effort. In the two previous experiments in the house-scene domain, where there is a comparable number of objects, the knowledge-engineering process took 36 man-months and $7\frac{1}{2}$ man-months, respectively. Here, we reduced the effort to 3 man-months using the tools that have been developed. This suggests that it is feasible to use this methodology to extend the knowledge base, and adapt it to other domains. Examples of road-scene interpretation results are shown in Figures 1-9.

2.3 The Schema Shell

The Schema Shell [15] is a software system implemented on TI Explorers that supports the development of large systems of schemas for image-interpretation tasks. Each schema contains knowledge about recognizing a class of objects. It has data declarations for collecting relevant information, and procedures (called *strategies*) for determining whether, when and how to ascertain that information. Parameterized instances of schemas are then invoked to interpret an image.

The Schema Shell provides mechanisms for interactively building schemas, and simulates a distributed environment (until parallel hardware arrives) in which an arbitrary number of schema instances may run concurrently. Schema instances communicate through a central blackboard. At any point during its processing a schema strategy may write an arbitrary message to the blackboard. Every other schema is then free to read, erase or modify that message. This provides for a single, uniform communication mechanism between schemas which can also be easily implemented on a variety of distributed architectures. The Schema Shell provides facilities for monitoring the behavior of schemas and their interactions during interpretation, allowing rapid, interactive development of schemas and associated strategies in a particular domain.

2.4 Intermediate Symbolic Representation

The input to high-level vision processes is *intermediate-level* data, which comes from two sources: the output from low-level processes such as line extraction and region segmentation; and the output from intermediate-level processes of grouping and selection.

Intermediate-level image descriptions are stored in the Intermediate Symbolic Representation, (or ISR). The ISR is a database which has been custom-built for the efficient storage, manipulation, and retrieval of abstract image data. The fundamental unit of representation is the *token*, each of which has a unique name, and a list of feature slots. The ISR can be used to store anything that can be fully characterized by a list of features and values; some of the image events currently stored include region segments, extracted edge lines, areas of homogenous texture, rectilinear line groups, and region-line relations. The benefits which result from imposing a uniform representation and user interface on all intermediate level tokens are enormous. It now becomes natural to think in terms of multistage and hierarchical grouping processes which take in tokens at one level of abstraction and produce tokens at the next higher level [27]. It also becomes more tractable to compare different types of tokens, which is necessary, for example, when relating edge lines to the regions they intersect [8]. Of course, the sharing of results and the elimination of data reformatting routines are obvious advantages.

At the highest level, tokens are partitioned according to the *image* they were extracted from. There is also an intermediate level of partitioning called the *tokenset*. Each feature associated with the tokens in a tokenset has a name, a value slot, a data type, and a computation function. Since most tokens have a physical realization in an image, a special bitplane data type is provided for representing the subset of image pixels which are associated with a token. Operators exist for taking the intersection, union, and difference of token bitplanes.

The ISR supports efficient access functions to tokens and sets of tokens; some of these access functions are associative in nature in that tokens may be accessed by means of constraints on feature values. In addition, the features may be precomputed or computed on a demand basis when the token/feature slot is accessed. The ISR allows a schema to create a token during an interpretation, create its bitplane either from scratch or as some combination of token bitplanes, and access its features, at which time new feature values will be automatically calculated for the new token. Thus, it is possible for schemas to dynamically "correct" misleading segmentations based on combinations of top-down knowledge.

The ISR has a consistent interface to both C and Common Lisp, permitting processes at all levels to access it in a uniform fashion. While not strictly part of the ISR, an Ethernet interface has been created between the TI Explorers and DEC VAXes, permitting schemas running on an Explorer to invoke low-level and intermediate-level processing on the VAXes. This allows a certain degree of parallel processing, and an appropriate division of labor between the various machines.

2.5 Object Hypothesis System

Bottom-up activation of schemas can be accomplished by forming initial object hypotheses on the basis of attributes of the initial image description expressed in terms of lines

and regions. Previously we have reported on rule-based approaches to initial hypothesis generation [8,18] which used a heuristic approach to forming constraints (rules) based on a theoretical Bayesian approach to maximum likelihood decisions over feature distributions. Recently, Lehrer and Reynolds [25] have extended the work and have developed a new object hypothesis system based on the Shafer-Dempster [14,29] theory of evidence. Their approach provides a more formal and theoretical foundation for the definition and interpretation of world knowledge. Object specific knowledge is defined automatically using statistical information obtained from a set of training object instances and a computationally efficient approach to the Dempster-Shafer theory of evidence is used for the representation and combination of evidence from multiple sources.

In this approach, the relationship between an object and its attributes is captured in a "plausibility" function. When applied to the primitive tokens (e.g. regions) the plausibility functions return evidence for or against an object hypothesis. The evidence from multiple plausibility functions is combined using an efficient computational algorithm to produce the final hypothesis. A large scale experiment is being planned for comparing and evaluating the results of the two systems.

2.6 Grouping and Perceptual Organization

We are initially viewing the task of perceptual organization and grouping as the extraction of relevant structure from overfragmented and incomplete descriptions and the construction of more abstract descriptions from less abstract ones. By this we mean algorithms which have as input the tokens produced by the low-level system and other grouping operations (region, lines, flow fields, ...) and have as output more complex tokens generated by grouping strategies based on the *relations* between the tokens. The goal of this type of 'intermediate' level processing is the reduction of the substantial representational gap which exists between the low level image descriptors and the primitives with which the high level semantic descriptions are constructed. The process of abstraction thus involves the search for events which can be more concisely described as a unit, and results in a description which may be more relevant to the evolving semantic interpretation.

Considerable progress has been made in developing grouping algorithms at the intermediate level of representation. The intermediate symbolic representation, described earlier, has been developed as the supporting representation for this work and several algorithms developed previously have been cast within this framework. A number of the local strategies for using the rank-ordered object hypotheses generated by the rule-structured initial object hypothesis system [18] can be viewed as grouping strategies. The extensions to this system developed by Belknap [9], which fuses information across multiple token types by means of relations expressed as rules, is also a form of grouping and has successfully generated object hypotheses from a combination of geometric and spectral features. Boldt [36,13] has developed a scale-sensitive hierarchical algorithm for grouping collinear line segments into progressively longer segments on the basis of geometric properties of the

hypothesized group as well as the similarity of image features along both sides of the component lines. A summary of these algorithms and a more comprehensive discussion of their relationship to perceptual organization and grouping may be found in [19].

As a result of these preliminary studies related to grouping, we [36,27] are developing a computational framework for geometric grouping and other organizational algorithms which addresses a set of overlapping issues. Clearly one must consider the extraction and representation of primitive tokens, the features of these tokens, and important relations between the tokens. In the case of geometric grouping algorithms this would include the extraction of lines and geometric relations such as collinearity, parallelness, relative angle, and spatial proximity derived from the Gestalt Laws of perceptual organization. One must also provide the means for expressing domain constraints in terms of these relations: i.e. grouping *strategies* must be defined and invoked based on knowledge of the domain and the current state of the system. Finally the system must deal explicitly with the problem of search, and its relation to the objects in the domain which are to be hypothesized and identified. In general each step of any grouping strategy must apply constraints which either significantly reduce the search space or add important information to the descriptive power of the system.

A number of algorithms are being developed at UMASS which satisfy these requirements and a computational framework has been proposed for confronting the issues described above. We view the grouping and search processes as part of a four-stage iterative grouping and extraction strategy which can be summarized as follows:

- *Primitive Structure Generation*: These processes provide the primitives (regions, lines, possibly surfaces, and in general, tokens) which are the input to the grouping and hypothesis generation process described next.
- *Linked Structure Generation*: This step applies very general geometric constraints to obtain graphs within which search processes can be applied to identify specific objects of interest. For example rectilinear structures which would contain rectangles or other simple geometric structures. This is essential for generating search spaces of reasonable size.
- *Subgraph Extraction*: This step involves the extraction of specific structures "one step up" the abstraction hierarchy, and uses the linked structures to constrain the search.
- *Replacement and Iteration*: Having extracted more abstract tokens, these can now play the role of primitives in another pass of grouping and extraction.

In [36,13] this strategy has been applied with striking results for the purpose of extracting straight lines. In [27] this strategy is being applied for the purpose of extracting rectilinear structures. In unpublished work, Lance Williams is developing an algorithm for

using a flow field generated from a motion sequence of images, to assist in the straight line extraction and temporal grouping process with excellent preliminary results.

2.7 Goal-Directed Intermediate-Level Executive

Kohl [22] has been developing a schema-based system called GOLDIE for intelligently controlling the application of parameterized low-level and intermediate-level processes on the basis of goals and constraints generated by the high-level interpretation system. Initially, GOLDIE (for Goal-Directed Intermediate Level Executive) was formulated as a goal-oriented resegmentation system which allowed top-down control over the low-level segmentation processes and this remains an important aspect of its function. However, it also has become clear that top-down control of the intermediate-level grouping processes is required; consequently GOLDIE has been extended to include these processes in its repertoire. Both the Boldt line grouping algorithm [13,36] and a rule-based region merging algorithm are incorporated into it, and we are examining further extensions. GOLDIE responds to requests from the interpretation processes through the goal structure by translating the goals into appropriate low-level and intermediate-level process specifications and then executing the process. The constraints imposed on the output of the process can be quite general; if the resulting structure does not satisfy the request, the system attempts to generate other strategies, using whatever contextual and semantic knowledge is available, in order to meet the constraints.

2.8 Low-Level Vision System

The new Low-Level Vision System (LLVS) is the successor to the VISIONS Image Operating System. LLVS allows efficient operation on pixel data for image processing and feature extraction in a multiresolution processing cone, under control of higher-level interpretation processes or through a convenient user interface. It also provides a substantial library of image operations. For compatibility with efforts at other institutions, LLVS is based on Common Lisp (for interactive user control) and C (for efficiency in low-level processing); it is currently implemented on VAXes.

3 Visual Motion Analysis

Our research in motion analysis has continued with a blend of theoretical and experimental investigations. There has been a concentration on the development of techniques that will find practical use in mobile vehicle navigation. In particular, we are in the process of transferring a motion algorithm from UMass to CMU for recovery of environmental depth under known motion; we expect it to be useful for both obstacle avoidance and landmark recognition. We now discuss those efforts directly supported by the ALV Project.

3.1 Recovery of Motion Parameters

Past motion research at UMass concentrated on the recovery of sensor motion parameters from analysis of two image frames obtained from a sensor in motion. Building on this research, Pavlin [26] has evaluated the Lawton algorithm for translational motion [23,24], and determined that the algorithm can be applied effectively with analysis of only 8 to 16 image points between frames if the sensor is pointed approximately in the direction of sensor motion. In addition, he has speeded up the algorithm and made it more robust by improving the optimization technique used for finding the translational focus of expansion (FOE). This was accomplished by computing the error measure for the assumed FOE from a sparse sampling of the visual field (or a more restricted area if constraints on the possible location of the FOE are available). Then, a smooth surface is fit to the error values at those points and the computed minimum of this surface is used to focus the search in the next step of an iterative optimization process.

Extending work done by Shariat and Price [30,31], Pavlin has also begun investigating the problem of determining fully the parameters of motion of a rigid body rotating with constant angular velocity while moving with constant translational velocity in space. His solution uses a similar optimization technique in a nine-dimensional parameter space. Promising initial results have been obtained using simulated data giving the image positions of a single object point over multiple frames. The chief issues at this stage seem to be the choice of starting point for the iterative optimization process, and the avoidance of spurious local optima.

3.2 Refinement of Depth Maps

Bharwani et al. [10,11,12] have continued to develop an algorithm that will compute increasingly more accurate depth information from a sequence of frames derived from approximately known translational motion of the sensor. This algorithm is intended to be applied after FOE recovery using the Lawton-Pavlin algorithm, or when vehicle instrumentation supplies sensor motion parameters. The algorithm matches points between frames up to some match resolution, computes a depth range for the environmental point, and then uses this information to predict a smaller search window in future frames, which then can be searched with finer match resolution and consequently more accurate depth. An important characteristic of this algorithm is that the temporal depth refinement can be applied at a constant computational rate and therefore is well-suited for robot navigation.

This algorithm has now been modified to take into account Snyder's theoretical treatment of uncertainty [33] discussed below. Because the positions of the FOE and of image features in the first frame are uncertain (because of digitization error and noise), the search for corresponding features in the second frame must be over appropriate two-dimensional regions. This modification has improved the robustness of the algorithm. In addition, the shape of the correlation surface [3] (whether it is peaked or flat) can be used to dynamically

control the limits of resolution of the match process.

3.3 Analysis of Frame-to-Frame Correspondence

Snyder [33,34], has theoretically examined the problem of uncertainty of image measurements in correspondence-based techniques, and their impact in stereo and motion analysis. The location of the FOE (as computed say by some motion algorithm) and of image features (as computed by an interest operators or by correlation matching) can be determined only approximately. At best, there is sub-pixel uncertainty ($\pm 1/2$ pixel) due to digitization. Uncertainty in such image locations leads to uncertainty in the recovery of depth from both stereo and motion, defines limits to the effectiveness of recovering depth of environmental points or detecting the presence of independently moving objects, and provides the means to determine the relative efficacy between stereo and motion analysis in varying situations. The analysis provides strategies for intelligently controlling the application of stereo and motion algorithms and determining uncertainty ranges for the results that are extracted.

Using a similar geometric setup, Smid et al. [32] have conducted a comparative evaluation of several commonly used interest-point operators, particularly as they might be used in techniques for processing translational sequences, like those described above. This evaluation will guide the selection and application of appropriate operators for such uses.

3.4 Software for Motion Analysis

The two algorithms mentioned above, for FOE recovery and temporal depth refinement, are being packaged into a motion-analysis software subsystem for transfer to CMU, and for use on the UMass mobile robot. The goal is the analysis of an ongoing sequence of frames from a vehicle in motion to determine obstacles in the path of motion. It is hoped that at CMU this subsystem will operate effectively at a range beyond the useful range of the ERIM sensor (40 foot limit).

There are three main stages of processing: First, frames must be registered since (at this time) the camera is not stabilized and therefore jerks, bumps, rocking, etc. will introduce local random translational and rotational motion between frames even when the vehicle is undergoing approximately pure global translation. Currently, we have a simple registration scheme involving the selection of distinctive points (high contrast and high curvature) that are at a great distance (near horizon), from which the rotational component can be estimated and subtracted out. Next, the FOE will be recovered by the Lawton-Pavlin algorithm using a small number of distinctive points, say 8, in the foreground (10-40 feet). Then the depth of distinctive points in the path of the vehicle will be computed using the Bharwani algorithm. Finally, either point sets that imply vertical surfaces, or individual points that are not consistent with lying on a planar road surface will be flagged for higher-level navigational attention.

The set of four programs for registration, FOE extraction, depth from motion, and obstacle detection has been tested on one real sequence of images obtained from the CMU NAVLAB. The programs are being ported to CMU for integration into their ALV effort, and further evaluation will take place at both UMass and CMU.

4 Mobile Robot Navigation

Vision-based mobile robot navigation is a relatively recent addition to the VISIONS research group at UMass. We have acquired a mobile robot that enables us to develop a testbed for techniques for dynamic image interpretation and for integrating these techniques into a complete system for autonomous robot navigation. The robot is to be operated both indoors and out, providing a wide variety of scenes for analysis. A fuller discussion of the mobile-robot project at UMass can be found in [7], and also in the reports [4,5,6].

Short-term goals of this project include the finalization of the depth-from-motion algorithm in a form that is useful for obstacle avoidance applications. This algorithm is in the process of being transferred to the Carnegie-Mellon University vehicle navigation project (see Motion Analysis section of this report). Our vehicle should be able to navigate cluttered hallways and sidewalks solely using visual data. Installation of a recently acquired UHF transmitter link should be completed soon, allowing the vehicle a greater range than it currently has in its tethered form.

4.1 Autonomous Robot Architecture

The UMass Autonomous Robot Architecture (AuRA) is being developed to support this research effort. It incorporates both global and reflexive schema-based path planning strategies and utilizes a priori knowledge stored in long-term memory, when available, to assist the vehicle's attainment of its navigational goals.

The chief navigational issues addressed include path following, landmark recognition for vehicle localization and obstacle avoidance. Path planning is handled at two levels. First, the computation of a global path is conducted based on information stored in long-term memory in the form of a meadow-map. An A* search algorithm capable of dealing with the multiple terrain types found in the map is used to determine the initial route. Then information contained within the map is used to provide appropriate motor behaviors (motor schemas) to enable the robot to attain its navigational goals. Multiple concurrent processes, developed only in simulation thus far, provide the velocity vectors that constrain the robot's motion. Motor schemas afford a relatively straightforward mechanism, using a potential field methodology, for the combination of the outputs of individual motor tasks. These can readily reflect the uncertainty of the perceived environmental objects. Examples of the potential fields generated by motor schemas are shown in Figures 10-11.

A hierarchical planning system consisting of a mission planner, navigator and pilot is being constructed to handle the task of path planning in both indoor and outdoor environments. Terrain features are taken into account in the determination of the best path for the mobile vehicle. The representations used will include a partial internal model of the environment. This enables the navigator to take advantage of a priori knowledge of the world while the pilot handles unanticipated and unmodeled obstacles as required.

Different path optimization strategies can be used based upon the mission's needs. Whether the safest path, shortest path, or some other metric constitutes the best path will depend on several factors. These would include the nature of the mission, the terrain to be traversed, temporal constraints, energy levels, positional uncertainty, etc. By modeling the free space of the vehicle's world expressly and tying relevant symbolic information to these "meadows", multiple factors are available for path-planning heuristics.

Possibly conflicting sensory input will have to be reconciled using "short-term memory" representations. The meadow map used for navigation will provide regions for instantiation based upon the robot's current position. Information from vision, ultrasonic sensors and positional sensors will be stored in this representation with associated certainty factors that will be altered based upon concurring or contradictory sensor input. This architecture will be sufficiently open-ended to allow the integration of additional sensor modalities (e.g. laser rangefinder, inertial guidance) as they become available.

Spatial and rotational uncertainty regarding the vehicle's position and orientation will be expressly modeled. The resulting spatial error map will be used to guide visual interpretation, windowing the image to reduce the time required for sensory processing. The sensory interpretations then will be used to reshape and reduce the spatial uncertainty map. The feedback provided by the sensors thus restricts the possible positions and orientations of the vehicle, while the probable location of the vehicle is used to guide sensory processing.

Homeostatic control (maintenance of the robot's own internal environment) is another research area. When mobile vehicles become capable of entering hazardous environments and covering longer distances without human monitoring, the status of the robot's energy levels, temperature, and other relevant variables can and should significantly affect planning and action. Through the use of internal sensors (in contrast to environmental sensors), surveillance of the internal state of the robot can be maintained. The information can then be used as necessary to change parameters for motor power consumption, heat production, etc., as well as provide data to the planner for decision making. Any vehicle purported to be "autonomous" must address this issue.

Many of the issues involved in the mobile vehicle research can be seen as complementary to those of other areas in our vision and robotics groups. The use of perceptual and motor schemas in the proposed vehicle architecture exploits many of the concepts used in both the VISIONS scene interpretation group and the work being done for the Laboratory for Perceptual Robotics distributed programming environment. Multi-sensor integration,

certainly crucial for the vehicle's domain, will only benefit from the work being done on the integration of vision, touch and force sensing.

4.2 Vision Modules

We have developed a number of vision modules, which, while of more general use, are specifically intended for use in robot navigation. The depth-from-motion algorithm developed by Bharwani, Hanson and Riseman [12] and discussed in Section 3.2 is nearly completed and will be used initially for obstacle avoidance. It can also provide information for landmark identification when coupled with top-down knowledge of expected landmark locations. Kahn, Kitchen and Riseman [20] have implemented techniques for extraction of straight lines and regions, techniques specifically tailored for fast execution in the context of robot navigation. As mentioned below, the fast line-extraction module has been used for path following, and will also be used for landmark recognition and vehicle localization. Examples of the operations of the fast line extractor on a typical image are shown in Figures 12-15. The more recently completed fast region-segmentation module has potential for the same applications. An analysis of edge operators conducted by Kitchen and Malin [21] provided guidance for selecting edge operators for the fast line extractor, and for setting its parameters for best performance. A description of all these algorithms and their use within AuRA can be found in [7].

We are exploiting fast hardware and parallel processing for speeding up robot navigation tasks. Our algorithms for line and region extraction are currently being adapted to use the pipeline image-processing capabilities and look-up tables of our Gould image processor. The AuRA system already uses concurrent processes running on several Vaxes to exploit parallelism for the path-following task. We expect to make use of a new Sequent multiprocessor to further decrease the processing time required for both vision and motor tasks and to enhance the real-time capabilities of the mobile robot project. When the UMass Image Understanding Architecture [35] is complete, much of the VISIONS system can be migrated directly into AuRA for real-time visual perception.

4.3 Robot Navigation Runs

The successes in actual robot experimentation to date are modest. Successful navigation of both an outdoor sidewalk and an indoor hall has been achieved. This has typically involved the robot's moving down the center of the path for distances of 30 feet in 5-foot stages, with about 45 seconds of computation between stages. Navigation has been based almost exclusively on visual guidance using the fast line-extraction routine. Dead-reckoning information is used minimally in our system, as our goal is to serve as a testbed for vision algorithms. The only use of dead reckoning is to provide approximate predictions of where in the image to apply the line-extraction routine. The algorithm is quite robust working with (unchanging) environments in the presence of significant path edge discontinuities

(doorways, vehicle tracks, clutter etc.), and poor contrast of the path with its surroundings (asphalt against grass in gray-scale images). An example of path extraction is shown in Figures 16-19. Obstacle avoidance on vehicle runs has been handled using ultrasonic data thus far.

References

- [1] G. Adiv, "Interpreting Optical Flow", Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, September 1985. Also COINS Technical Report 85-35.
- [2] P. Anandan, "Measuring Visual Motion From Image Sequences", Ph.D. Dissertation, University of Massachusetts at Amherst, 1987.
- [3] P. Anandan, "A Unified Perspective on Computational Methods for the Measurement of Visual Motion", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [4] R. C. Arkin, "Path planning and execution for a mobile robot: A review of representation and control strategies", COINS Technical Report 86-47, Computer and Information Science Department, University of Massachusetts at Amherst, October 1986.
- [5] R. C. Arkin, "Path planning for a vision-based mobile robot", to appear in *MOBILE ROBOTS — SPIE Proc.*, Vol. 727, 1986. Also COINS Technical Report 86-48, Computer and Information Science Department, University of Massachusetts at Amherst, October 1986.
- [6] R. C. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior", to appear in *Proc. IEEE International Conference on Robotics and Automation*, Raleigh, N.C., 1987.
- [7] R. C. Arkin, E. M. Riseman, and A. R. Hanson, "AuRA: An Architecture for Vision-Based Robot Navigation", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [8] R. Belknap, E. Riseman, and A. Hanson, "The Information Fusion Problem and Rule-Based Hypotheses Applied to Complex Aggregations of Image Events", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, FL, June 22-26, 1986, pp. 227-234.
- [9] R. Belknap, E. Riseman, and A. Hanson, "The Information Fusion Problem and Rule-Based Hypotheses Applied To Complex Aggregations of Image Events", COINS Technical Report, University of Massachusetts at Amherst, in preparation, 1987.

- [10] S. Bharwani, E. Riseman, and A. Hanson, "Refinement of Environmental Depth Maps over Multiple Frames", *Proc. of the DARPA Image Understanding Workshop*, Miami Beach, FL, December 1985.
- [11] S. Bharwani, E. Riseman, and A. Hanson, "Refinement of Environmental Depth Maps over Multiple Frames", *Proceedings of the IEEE Workshop on Motion: Representation and Analysis*, Charleston, SC, May 7-9, 1986, pp. 73-80.
- [12] S. Bharwani, E. Riseman, and A. Hanson, "Multiframe Computation of Accurate Depth Maps Using Uncertainty Analysis", forthcoming technical report, Computer and Information Science Department, University of Massachusetts at Amherst.
- [13] M. Boldt and R. Weiss, "Geometric Grouping Applied to Straight Lines", forthcoming technical report, Computer and Information Science Department, University of Massachusetts at Amherst.
- [14] A. P. Dempster, "A Generalization of Bayesian Inference", *Journal of the Royal Statistical Society, Series B*, Vol. 30, 1968, pp. 205-247.
- [15] B. Draper, R. Collins, J. Brolio, J. Griffith, A. Hanson, and E. Riseman, "Tools and Experiments in the Knowledge-Based Interpretation of Road Scenes", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [16] F. Glazer, "Hierarchical Motion Detection", Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, 1987.
- [17] F. Glazer, "Hierarchical Gradient-Based Motion Detection", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [18] A. Hanson and E. Riseman, "A Methodology for the Development of General Knowledge-Based Vision Systems", to appear in *Vision, Brain, and Cooperative Computation*, (M. Arbib and A. Hanson, Eds.), 1987, MIT Press Cambridge, MA. Also COINS Technical Report 86-27, University of Massachusetts at Amherst, July 1986.
- [19] A. Hanson and E. Riseman, "The VISIONS Image Understanding System — 1986", in *Advances in Computer Vision*, (Chris Brown, Ed.), Erlbaum Press, 1987.
- [20] P. Kahn, L. Kitchen and E. Riseman, COINS Technical Report, University of Massachusetts at Amherst, in preparation.
- [21] L. Kitchen and J. Malin, "The effect of spatial discretization on the magnitude and direction response of simple differential edge operations on a step edge. Part 1: square pixel receptive fields", Technical Report 87-34, Computer and Information Science Department, University of Massachusetts at Amherst, April 1987.

- [22] C. Kohl, A. Hanson, and E. Riseman, "A Goal-Directed Intermediate Level Executive for Image Interpretation", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [23] D. T. Lawton, "Processing Translational Motion Sequences", *Computer Graphics and Image Processing*, Vol. 22, pp. 116-144, 1983.
- [24] D. T. Lawton, "Processing Dynamic Image Sequences from a Moving Sensor", Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, 1984. Also COINS Technical Report 84-05.
- [25] N. Lehrer, G. Reynolds, and J. Griffith, "Initial Hypothesis Formation in Image Understanding Using an Automatically Generated Knowledge Base", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [26] I. Pavlin, E. Riseman, and A. Hanson, "A Translational Motion Algorithm Using Hierarchical Search with Smoothing", forthcoming technical report, Computer and Information Science Department, University of Massachusetts at Amherst.
- [27] G. Reynolds and R. Beveridge, "Searching for Geometric Structure in Images of Natural Scenes", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [28] E. M. Riseman and A. R. Hanson, "A methodology for the development of general knowledge-based vision systems", *IEEE Workshop on Principles of Knowledge-Based Systems*, Denver, Colorado, December 1984, pp. 159-170.
- [29] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
- [30] H. Shariat, "The motion problem: How to use more than two frames", Ph.D. Dissertation, University of Southern California, Los Angeles (Tech. Report IRIS 202, Instit. for Robotics and Intelligent Systems, Oct. 1986).
- [31] H. Shariat and K. E. Price, "How to use more than two frames to estimate motion", *Proceedings Workshop on Motion: Representation and Analysis*, Charleston, South Carolina, May 7-9, 1986, pp. 119-124.
- [32] F. Smid, L. J. Kitchen and E. M. Riseman, "A study of interest-point operators for token-based image matching", forthcoming technical report, Computer and Information Science Department, University of Massachusetts at Amherst.
- [33] M. Snyder, "Uncertainty Analysis in Image Measurements", forthcoming technical report, Computer and Information Science Department, University of Massachusetts at Amherst.

- [34] M. Snyder, "Uncertainty Analysis in Image Measurements", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [35] C. Weems, S. Levitan, E. Riseman, and A. Hanson, "The Image Understanding Architecture", *Proceedings of the DARPA Image Understanding Workshop*, Los Angeles, CA, January 1987.
- [36] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, FL, June 22-26, 1986, pp. 489-495.
- [37] T.E. Weymouth, "Using Object Descriptions in a Schema Network For Machine Vision", Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst. Also COINS Technical Report 86-24, University of Massachusetts at Amherst, 1986.

Figures



Figure 1. Sample road-scene photograph.

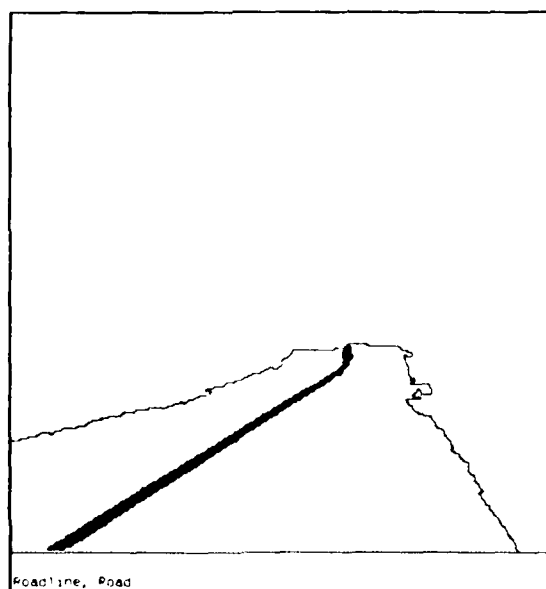


Figure 2. Interpretation results for Figure 1.
Road line: black;
Road: light grey.



Figure 3. Interpretation results for Figure 1.
Tree trunk: black;
Gravel: dark grey;
Foliage: light grey.



Figure 4: Road scene with an uneven ground plane.

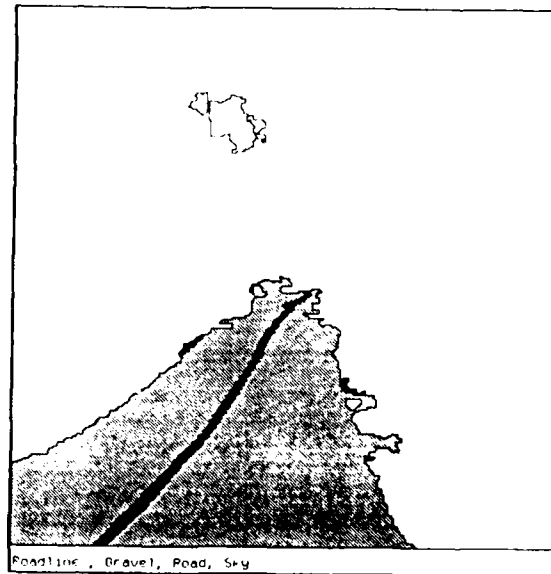


Figure 5. Interpretation results for Figure 4.

Road line: black;
Gravel: dark grey;
Road: medium grey;
Sky: light grey.

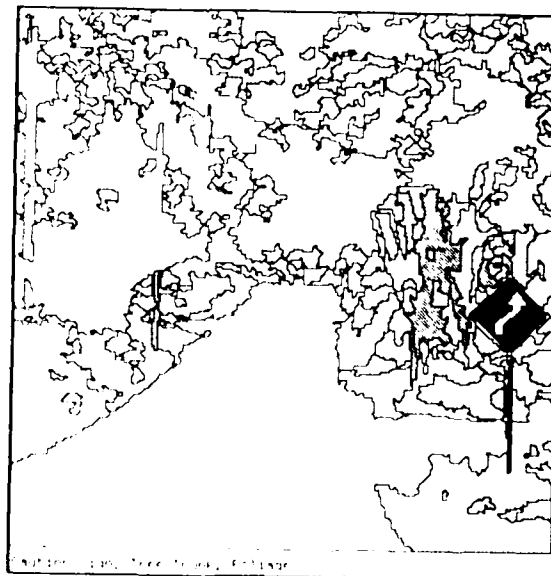


Figure 6. Interpretation results for Figure 4.

Caution sign: black;
Tree trunk: medium grey;
Foliage: light grey.



Figure 7: Road scene with other man-made structures.

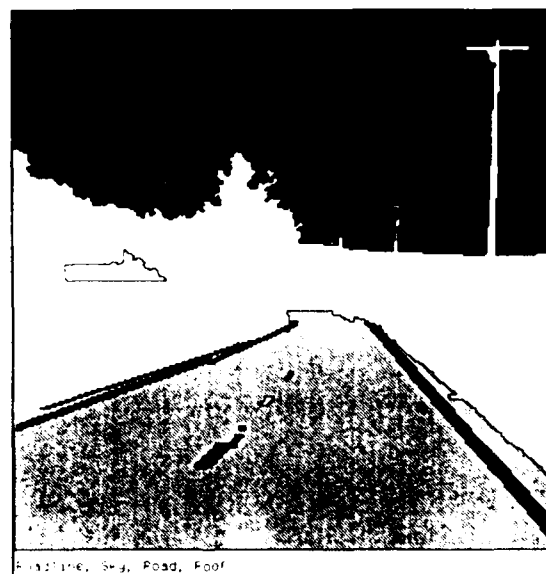


Figure 8. Interpretation results for Figure 7.

Road-line: black;
 Sky: dark grey;
 Road: medium grey;
 Roof: light grey.

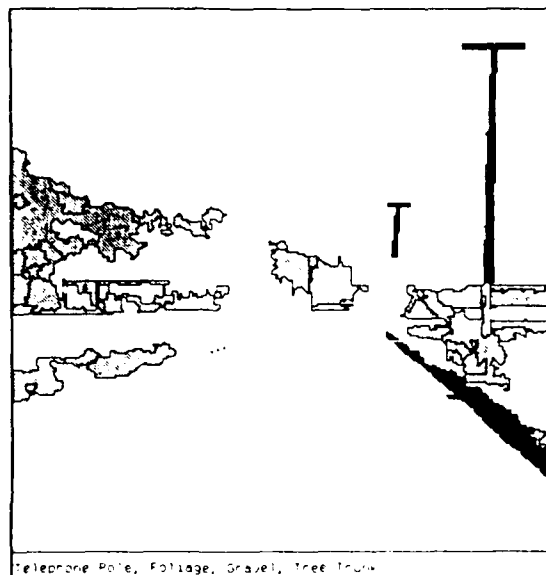


Figure 9. Interpretation results for Figure 7.

Road line: black;
Gravel: dark grey;
Road: medium grey;
Sky: light grey.

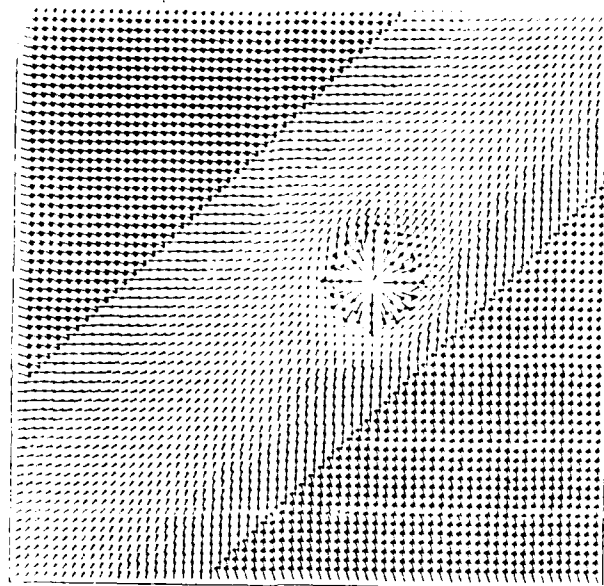


Figure 10. Potential fields produced by motor schemas during leg traversal. Before the goal is identified, the **move-ahead** and **stay-on-path** schema instances direct and constrain the robot on its way. A single **obstacle** schema instance is present. (The arrows represent the desired velocity vectors that constrain the robot's motion).

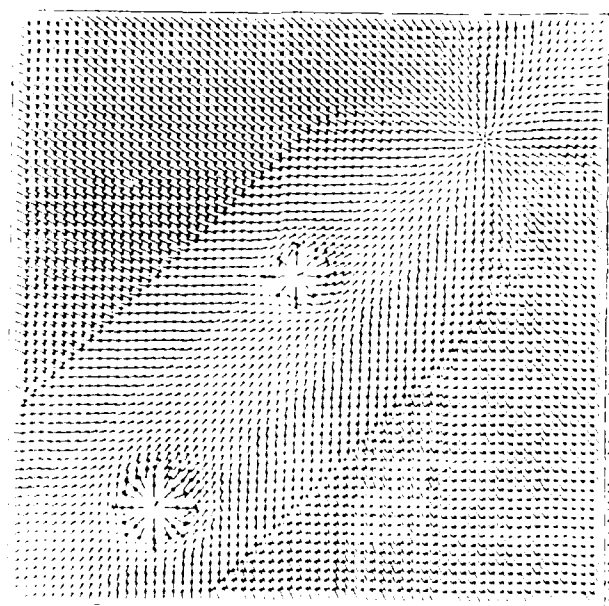


Figure 11. Potential fields produced by motor schemas during leg traversal. After the goal is identified, the **move-to-goal** schema instance replaces the **move-ahead** schema instance. Two **obstacle** schema instances are shown as the goal is approached. (The arrows represent the desired velocity vectors that constrain the robot's motion.)



Figure 15. Fast line finder. Lines extracted with parameter settings tuned to extract long vertical and horizontal lines above the horizon (typical of buildings and other landmarks).



Figure 16. Path extraction using fast line finder. Output of fast line finder run on image in Figure 12, with buckets tuned to find sidewalk edges (same as Figure 14).

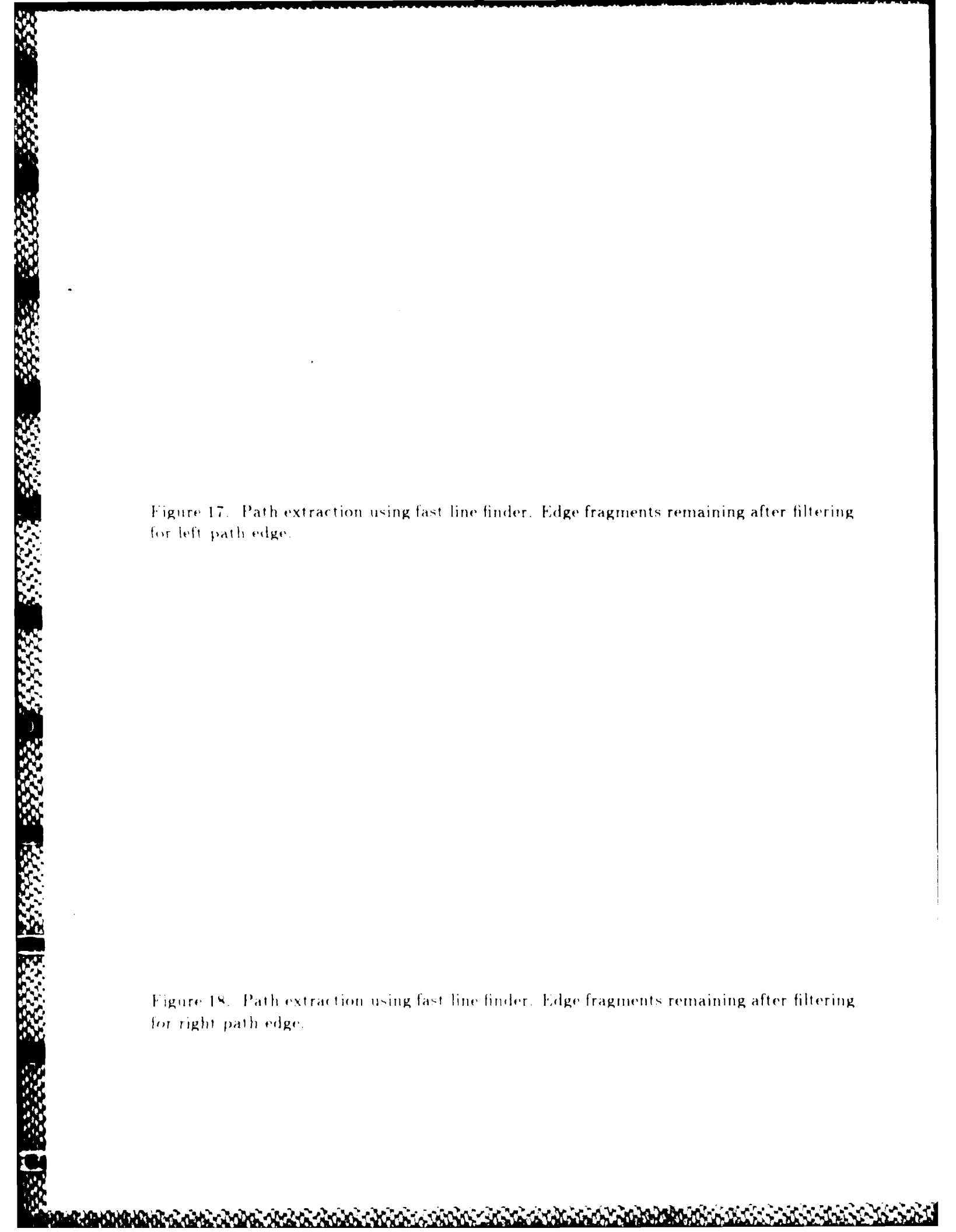


Figure 17. Path extraction using fast line finder. Edge fragments remaining after filtering for left path edge.

Figure 18. Path extraction using fast line finder. Edge fragments remaining after filtering for right path edge.

Figure 19. Path extraction using fast line finder. Resultant entire path boundaries and computed center line.

END

10-87

DTIC